

YANG Configuration Templates

draft-tt-netmod-yang-config-templates-02

Robert Wills (Cisco)

Qiufang Ma (Huawei, presenting)

Deepak Rajaram (Nokia)

Recap

- Two virtual interims to discuss the template requirements collected at: <https://github.com/netmod-wg/template-reqs/issues>
 - ~20 requirements reviewed & discussed, show-of-hands polling decides valid ones
 - Additional requirements submitted post-interims
- Submit as I-D.tt-netmod-yang-config-templates to document the outcomes of interims
- Solution Overview
 - A **Configuration Template** is a fragment of configuration in <running>
 - A node in <running> can **apply** a template to expand its configuration in <intended>
 - Config from templates can be **overridden** by explicitly provided regular config

Changes in -02

- New Section 4.5: Deletion of templates
- New Section 8: Operational considerations
- New Appendix A: Requirements implementation status

Top-priority open issues:

- MUST template be valid when defined?
- How to model the contents of a template?

Requirement Implementation Status

Requirement	Fulfilled	Comments from authors	Template-reqs Repo issue
R1: Allowed Multiple templates to be applied at a single node	Y	see Section 4.2	Requirement #1
R2: Templates must work with any YANG module	Y	see Section 4.1	Requirement #3
R3: Templates must be validated when defined	N	Needs further discussion, see Editor's note from Section 4.1	Requirement #4
R4: Local-config overrides template-config	Y	see Section 4.3	Requirement #6
R5: Living template: modified template data gets expanded for all consumers	Y	see Section 4.4	Requirement #7
R6: Support basic programmatic elements in templates	N	Seems to add some complexity	Requirement #8
R7: Allow a server to constrain which nodes can be templates consumer	Y	See Section 8	Requirement #9
R8: Configuration with both expanded and unexpanded templates is able to be returned	Y	see Section 5 and Section 8	Requirement #10

Notes: Requirements that is "Mostly Opposed" in the polling have not been taken into account, nor have editorial requirements.

Requirement Implementation Status (cont.)

Requirement	Fulfilled	Comments from authors	Template-reqs Repo issue
R9: <running> contains the unexpanded template	Y	see Section 5 , also stated explicitly in Section 8	Requirement #12
R10: <intended> contains the expanded template	Y	see Section 5 , also stated explicitly in Section 8	Requirement #13
R11: Enables off-box template expansion of <running>	Y	see Section 4.4	Requirement #14
R12: Support limited regex in templates	Y	But needs more work, see Section 4.1.1	Requirement #18
R13: Have a precedence rule when multiple templates are applied at a single node	Y	See Section 4.4	Requirement #19
R14: The innermost template takes precedence when templates are applied at multiple ancestor nodes	Y	See Section 4.4	Requirement #20
R15: Enable non-NMDA servers to return the expanded data	N	have a dedicated section (Section 6) for this, but empty now	Requirement #21
Not discussed: R16: exclude templates applied at ancestor nodes	N	Seems to add some complexity, needs further discussion	Requirement #22
Not discussed: R17: Annotations to determine which template a node was applied from	N	Needs further discussion	Requirement #24

Notes: Requirements that is “Mostly Opposed” in the polling have not been taken into account, nor have editorial requirements.

Open Discussion: MUST template be valid when defined?

The show-of-hands poll at the interim shows a split opinion.

- Unsure if “no” because opposed to the entire idea or to doing it in a first release.

The current draft states:

- *At definition time:*
 - Template content does not have to be fully valid
 - e.g., mandatory nodes can be missing
 - Where validation is possible (e.g., range, pattern, length), the server SHOULD enforce these constraints in the template
- *At expansion time:*
 - Config after template expansion MUST always be valid.

Is this acceptable?

Open Discussion: How to model the contents of a template?

- Template config is modeled as anydata in the current draft

```
module: ietf-config-template
  +--rw templates
    +--rw template* [id]
      +--rw id          string
      +--rw description? string
      +--rw content?   <anydata>
      +--ro last-modified? yang:timestamp
```

```
<templates xmlns="...">
  <template>
    <id>interface-mtu</id>
    <content>
      <interfaces xmlns="...">
        <interface>
          <name>eth*</name>
          <mtu>1500</mtu>
        </interface>
      </interfaces>
    </content>
  </template>
</templates>
```

The template contains a fragment of config using the ietf-interfaces model

- Limitations of using anydata:
 - No schema to declare what a fully valid template is
 - Again, should we mandate template validation at definition time?
 - Opaque data – not possible for the client to edit part of the template
 - See Sec.7.10.3 in 7950, any delta change to the template requires the client to send the entire template config

Alternative Option: Automatically derive a schema

- Model <content> as a presence container
- For each YANG model that the server supports, derive a second “template model” that augments /templates/template/content
- For example:
 - the server implements ietf-interfaces
 - the server derives a second model that augments /templates/template/content
 - To a client, the template content is just another YANG container whose contents is all the configuration that could live in that template
- Does the value of this option deserves such an implementation complexity?

Comments, Questions, Concerns?